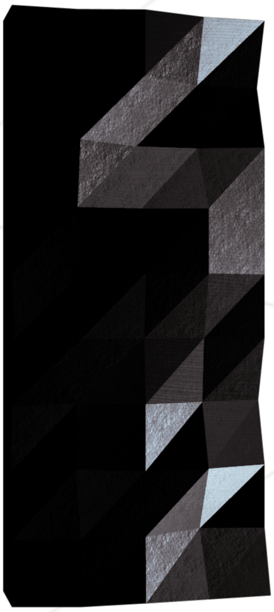


Rush 01 – Piscine Python for Data Science

Food and nutrition

Summary: This rush will help you to strengthen the skills acquired in the previous days

Contents

Chapter I

Foreword

You are what you eat. Think about your body and organism as a system that needs to get many different supplements to live happily and effectively. If it lacks some nutrients, important reactions may stop happening and your health will eventually be weak.

Your diet should be balanced if you want to live healthily. You need proteins, carbohydrates,and fats as well as Fe, Mg, Na, K, Ca, Zn, Se, Cr, I, vitamins D, B12, E, C, A, K, Cu, etc. It is already a long list but it is far away to be exhausted. If you do not eat enough proteins, the organism will start looking for them inside itself in muscles, organs, tissues. Eventually, you will lack enzymes, hormones, transport proteins, immune cells, etc. If you eat too much of them, the organism will be intoxicated by decay products and the health will decrease too.It applies to any item in the list. "Poison is in everything, and no thing is without poison. The dosage makes it either a poison or a remedy".

The first problem is that our diet is not as diverse as it is required if we talk about an average person in the Western world. How often do you eat vegetables, fruits, nuts, beans, berries, bread, pastries, fish, meat, seaweed, butter? How often do you drink coffee, green tea, milk, alcohol, juices, smoothies?

The second problem is that we like to eat something tasty and all this healthy food is not as tasty as fried fries, chips, etc. The third problem is that we do not know good recipes.

What if we had an app that could predict how tasty a dish can be with the current products from our refrigerator, what good nutrients it contains and recommend us a menu with healthy and tasty recipes at the same time?

Chapter II

Instructions

- Use this page as the only reference. Do not listen to any rumors and speculations about how to prepare your solution.
- Here and further we use Python 3 as the only correct version of Python.
- The python files for python exercises (module01, module02, module03) must have a block in the end: if ___name___ == '___main___'.
- Pay attention to the permissions of your files and directories.
- To be assessed your solution must be in your GIT repository.
- Your solutions will be evaluated by your piscine mates.
- You should not leave in your directory any other file than those explicitly specified by the exercise instructions. It is recommended that you modify your .gitignore to avoid accidents.
- When you need to get precise output in your programs, it is forbidden to display a precalculated output instead of performing the exercise correctly.
- Have a question? Ask your neighbor on the right. Otherwise, try with your neighbor on the left.
- Your reference manual: mates / Internet / Google.
- You can ask questions in Slack.
- Read the examples carefully. They may require things that are not otherwise specified in the subject.
- And may the Force be with you!

Chapter III

Specific instructions of the day

- No code in the global scope of the main program. Use the classes and methods that you have written in the development stage!
- Any exception not caught will invalidate the work, even in the event of an error that was asked you to test.
- The interpreter to use is Python 3
- Any built-in function is allowed

Chapter IV

Mandatory part

In this rush, you are going to work on your own application that will help people to eat healthier and tastier food. It is a good and valuable experience to create a product prototype out of different technologies that you have studied. That will be a nice result for just two weeks of diving in Python and data science!

Your work will have three stages: research, development, and the program itself. Each of the stages will produce corresponding files as a result.

Main program

Let us start from the end. The program is a Python script (nutritionist.py).

- It takes in a list of ingredients.
- It forecasts and returns the rating class (bad, so-so, great) of a potential dish with the ingredients.
- It finds and returns all the nutrients (proteins, fats, sodium, etc.) in the ingredients as well as their daily values in %.
- It finds 3 the most similar recipes to the list of ingredients, their ratings, and the URLs where a user can find the full details.

The example is below:

```
$ ./nutritionist.py milk, honey, jam
I. OUR FORECAST
You might find it tasty, but in our opinion, it is a bad idea to have a dish with that list of ingredients
     .

II. NUTRITION FACTS
Milk
Protein - 6% of Daily Value
Total Carbohydrate - 1% of Daily Value
Total Fat - 1% of Daily Value
Calcium - 12% of Daily Value
...

Honey
...
```

```
Jam
...

III. TOP-3 SIMILAR RECIPES:
- Sweet Buttermilk Spoon Breads, rating: 1.875, URL: https://www.epicurious.com/recipes/food/views/sweet-
    buttermilk-spoon-breads-351045
- ...
- ...
```

Development

You need to create a Python module (recipes.py) with the classes and methods that are used in the main script.

Class Forecast, NutritionFacts, SimilarRecipes can be found in attachments

Research

In this part of the rush, you need to prepare everything that is used in the classes and methods above in a Jupyter Notebook (recipes.ipynb).

- Forecast
    - Data preparation
        - Use the dataset from Epicurious collected by HugoDarwood.
        - Filter the columns: the less non-ingredient columns in your dataset the better. You will predict the rating or rating category using only the ingredients and nothing else.
    - Regression
        - Try different algorithms and their hyperparameters for rating prediction. Choose the best on cross-validation and find the score (RMSE) on the test subsample.
        - Try different ensembles and their hyperparameters. Choose the best on cross-validation and find the score on the test subsample.
        - Calculate the RMSE for a naive regressor that predicts the average rating.
    - Classification
        - Binarize the target column by rounding the ratings to the closest integer. It will be your classes.
        - Try different algorithms and their hyperparameters for class prediction. Choose the best on cross-validation and find the score (accuracy) on the test subsample.

    - ∗ Compare the metrics using accuracy. Calculate the accuracy of a naive classificator that predicts the most common class.
    - ∗ Binarize again the target column by converting the integers to classes ‘bad’ (0, 1), ‘so-so’ (2, 3), ‘great’ (4, 5).
    - ∗ Try different algorithms and their hyperparameters for class prediction. Choose the best on cross-validation and find the score on the test subsample.
    - ∗ Compare the metrics using accuracy. Calculate the accuracy of a naive classificator that predicts the most common class.
    - ∗ What is worse: to predict a bad rating which is good in real life, or to predict a good rating which is bad in real life? Replace accuracy with the appropriate metric.
    - ∗ Try different algorithms and their hyperparameters for class prediction with the new metric. Choose the best and find the score on the test subsample.
    - ∗ Try different ensembles and their hyperparameters. Choose the best and find the score on the test subsample.

  - ○ Decision

    - ∗ Decide what is better to use: the regression model or the classification. Save the best model. You will use it in the program.

- Nutrition Facts

  - ○ Collect all the nutrition facts for the ingredients from your prepared and filtered dataset (only ingredient columns) into a dataframe. Use the following API for that.
  - ○ Transform all the values into % of the daily value. Keep only ingredients that exist in this and that table.
  - ○ Save the transformed dataframe into a CSV file that you will use in your main program

- Similar Recipes

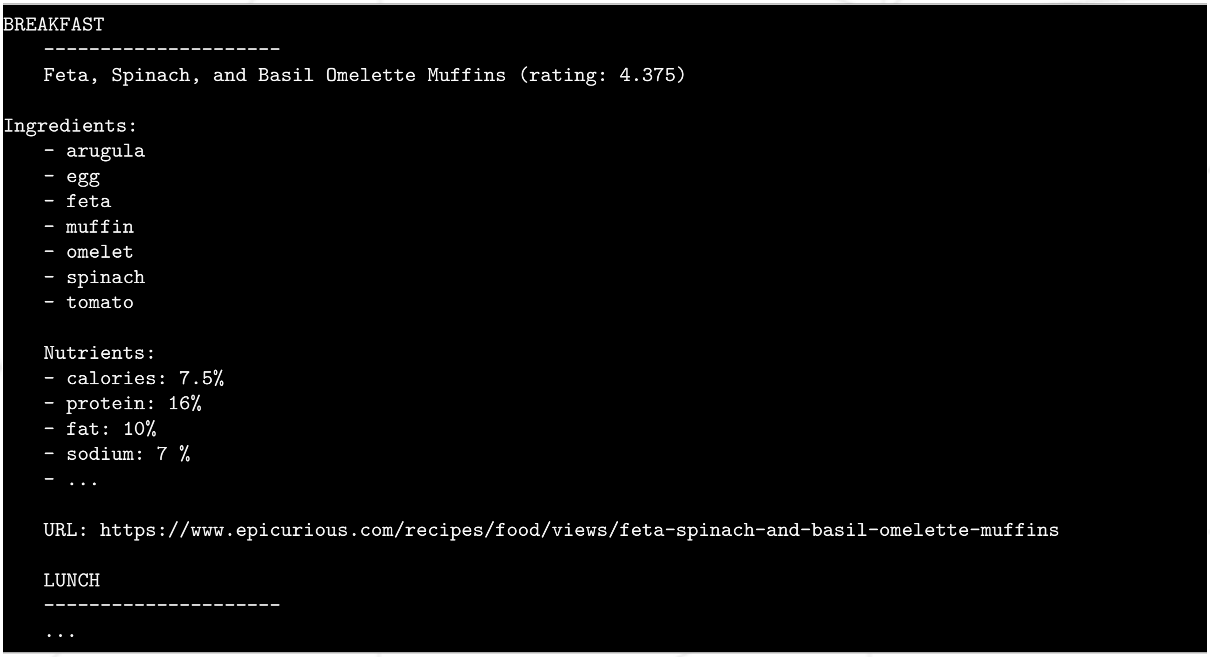
  - ○ For each recipe from the dataset, collect the URL from epicurious.com with the details (if there is no URL for that recipe, find a similar on the Internet).
  - ○ Save the new dataframe to a CSV file that you will use in your main program.

Chapter V

Bonus part

Add to the classes more methods that will help the script perform a new function: generate a menu for a day.

The day menu should randomly give a list of the three recipes that cover most of the needs in nutrients (% of the daily value) without overtaking them and have the highest total rating.
You should offer only appropriate recipes for breakfast, lunch, and dinner.
The result of the program should be like this:



```
BREAKFAST
   ---------------------
   Feta, Spinach, and Basil Omelette Muffins (rating: 4.375)

Ingredients:
   - arugula
   - egg
   - feta
   - muffin
   - omelet
   - spinach
   - tomato

   Nutrients:
   - calories: 7.5%
   - protein: 16%
   - fat: 10%
   - sodium: 7 %
   - ...

   URL: https://www.epicurious.com/recipes/food/views/feta-spinach-and-basil-omelette-muffins

   LUNCH
   ---------------------
   ...
```

Chapter VI

Turn-in and peer-evaluation

Turn in your work using your git repository, as usual.
Only the work that's in your repository will be graded during the evaluation.

During the correction, you will be evaluated on your turn-in as well
as your ability to present, explain, and justify your choices.